УДК 519.685

И. П. ГАМАЮН, д-р техн. наук,
С. В. ДРЕВАЛЬ, студент НТУ «ХПИ»

МОДЕЛИРОВАНИЕ ПРОЦЕССА ФУНКЦИОНИРОВАНИЯ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статті пропонується імітаційна модель процесу функціонування ліцензійного програмного забезпечення, на основі результатів якого можливо оцінити яке навантаження подолає ліцензійний сервер, враховуючи ресурсні витрати на обробку кожнозо запиту клієнта.

В статье предлагается имитационная модель процесса функционирования лицензионного программного обеспечения, на основе результатов которого можно оценить с какой нагрузкой справится лицензионный сервер, учитывая ресурсные затраты на обработку каждого запроса клиента.

In the article there have been proposed imitating modeling process of functioning of the license software on the basis of which results it is possible to estimate with what loading the license server will consult, considering resource expenses for processing of each inquiry of the client.

Постановка задачи исследования. В настоящее время одним из наиболее широко распространенных средств исследования и оптимизации функционирования сложных систем является имитационное моделирование, реализуемое с помощью современной вычислительной техники. Имитационная модель воспроизводит процесс функционирования исследуемой системы с учетом влияния внешней среды. Поэтому с помощью имитационной модели можно отрабатывать воздействия различных факторов, влияющих на поведение системы, изучать влияние изменения внутренних параметров на эффективность функционирования и так далее [1 – 4].

В данной статье предлагается имитационная модель процесса функционирования лицензионного программного обеспечения, разработанного на основе системы GPSS World. Это касается, прежде всего, программ, работающих в компьютерных сетях. Модель позволяет дать рекомендации для разработки единого центра проверки лицензий клиентских программ. В статье рассматриваются общие принципы построения взаимодействия между лицензионным сервером и программой, пытающейся проверить работоспособность своей лицензии. На основе сформулированных принципов было разработано тестовое программное обеспечение, включающее в себя лицензионный сервер и клиентскую часть. Затем на основе статистических данных была построена имитационная модель в GPSS World. Это позволило исследовать поведение системы для различных значений параметров и позволило определить направления их улучшения.

Имитационная модель. В сетевых и многопользовательских системах предполагается наличие лицензионного сервера. Лицензионная программа

проверяет на наличие ключевого диска или ключа и получает разрешение на исполнение от серверного процесса. Серверные процессы обеспечивают контроль за допустимым количеством пользователей программы, за тем, когда будет использоваться программа, где будет использоваться программа и даже за тем как будет использоваться программа. Поскольку большинство компьютеров работают в Internet, то метод контролируемого доступа к программному обеспечению и данным становится все более актуальным.

Для тестирования модели была разработана опытная система. В общем случае она работает следующим образом:

- пользователь (U) стартует лицензионную программу P;
- программа P обращается к серверу S за разрешением на запуск;
- сервер S проверяет текущее число пользователей, которые уже работают с программой P;
- если допустимый предел на количество пользователей еще не достигнут, то сервер S разрешает доступ и программа P начинает исполняться;
- если же предел по количеству пользователей был достигнут, то сервер S не дает разрешения на запуск, и программа P сообщает пользователю U, что нужно попытаться выполнить ее запуск позже.

Предлагаются программы “клиента” и “сервера” образующие систему. Сервер предоставляет разрешение прикладной программе на ее запуск, а также отслеживает выполнение лицензионных требований. Если лицензионный сервер не работает, то прикладная программа не сможет получить разрешение на запуск и откажет пользователю в запуске.

Лицензионный сервер на запросы выдает электронные билеты, которые содержат определенную комбинацию кодов. При этом нужно учесть, что случайно или преднамеренно серверу будут пытаться выслать заведомо неправильные пакеты данных. Для того, чтобы сервер не перегружался из-за неправильных сообщений, используется контрольная сумма всего сообщения. Если сумма не совпадает, то разрывается соединение с клиентом.

Схема алгоритма работы сервера показывается на рис. 1. Из алгоритма следует, что на сервере в самом начале организуется цикл, в котором происходит обработка поступающих запросов от клиента. Если очередной пакет распознан как данные от клиента, то начинается взаимодействие с ним, а в противном случае происходит повторное ожидание пакетов. Затем производится идентификация сообщения. Во время нее из сообщения извлекается информация, на основе которой производится обработка. Если пакет является запросом на освобождение клиентской лицензии, то из него извлекается ключ, идентифицирующий пользователя, и при помощи ключа производится удаление соответствующей записи в клиентской базе данных.

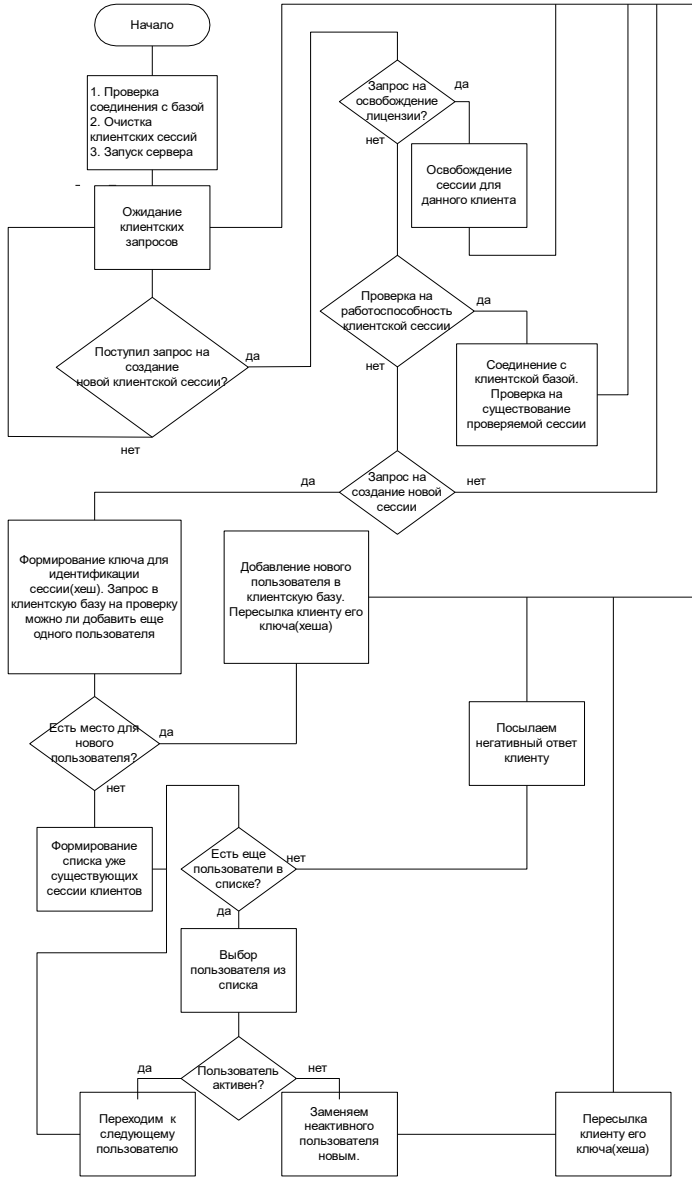

Рисунок 1 – Схема алгоритма работы сервера

Для проверки работоспособности клиентской сессии и прикрепленной к ней лицензии из сообщения извлекается ключ, по которому производится проверка существования записи. Положительный или отрицательный результат сообщается клиенту. При отрицательном результате клиент прекращает свою работу, так как считается, что у него нет лицензии. Этот случай возможен, когда кто-то ввел заведомо неправильный ключ или если сервер был перезапущен и текущая клиентская база утеряна. Пакет может представлять собой запрос на создание новой сессии для клиента, связанной с лицензией. При этом если все сессии, предусмотренные лицензией, уже заняты, то производится поиск клиентов, которые не активны, но свою сессию в базе имеют. В этом случае происходит замена такого клиента на нового. В любом случае, если клиент добавляется в базу, то для него создается уникальный ключ, который связывает клиента с его лицензией. Это необходимо для безопасности, так как даже если злоумышленник получит ключ, то он не сможет им воспользоваться, потому что они каждый раз меняются.

Для хранения журнала с записями лицензий и связанных с ними клиентов использовалось СУБД (система управления базами данных) на основе MySQL. Эта СУБД зарекомендовала себя как быстрая и стабильная, что важно для обработки большого количества записей. Также важно еще то, что она является мультиплатформенной системой и может работать в Windows и Unix системах.

Для формирования пакетов использовалось некоторое подобие формата Modbus RTU. Modbus — коммуникационный протокол, основанный на клиент-серверной архитектуре [5].

Создание имитационной модели на основе GPSS предшествует разработке тестовой системы. Как следует из алгоритма работы лицензионного сервера, она разделяется на две составляющие: взаимодействие с клиентами и работа с клиентской базой данных. При этом необходимо учитывать такую возможность как блокировки БД (база данных) при обращении к ней со стороны лицензионного сервера. Это необходимо для корректного создания списка пользователей при проверке их активности, а также других операций с пользовательскими сессиями. Необходимость блокировки БД учитывается в модели и поэтому для оценки времени необходимо вычислить отдельно временные затраты для взаимодействий с клиентами, на операции, которые не зависят от других открытых соединений для остальных клиентов и работы с БД, для которой происходит блокировка, касающаяся всех клиентов, подключенных в данный момент к серверу. Для оценки времени операций по взаимодействию с клиентскими приложениями и работы с БД разделим в соответствии с алгоритмом работы программы временной промежуток каждой операции на две части: работа сервера с клиентом независимо от других клиентов и время общей для всех блокировки БД при работе с ее записями. В программе можно выделить две основные

части, временные затраты которых будут наиболее существенны, а остальными можно пренебречь, поскольку они обрабатываются мгновенно и компьютер не способен зафиксировать время их выполнения. Этими частями являются: запрос на выделение новой клиентской сессии и запрос на проверку работоспособности клиентской сессии. При этом, если клиент запрашивает для себя новую сессию и свободных мест нет, то работа с таким клиентом заканчивается. Поэтому эту часть необходимо поделить еще пополам. В итоге мы имеем три части, в которых происходит взаимодействие с клиентом и БД. Для каждой были произведены замеры временных затрат, при нагрузке на лицензионный сервер. Было расчитано отдельно время, отводимое для взаимодействия с клиентом и БД.

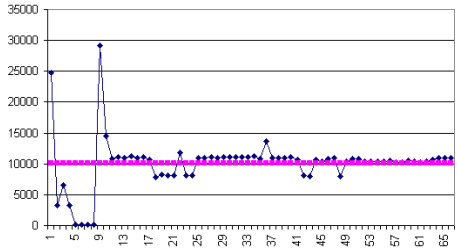

Рисунок 2 – Временные затраты на взаимодействие с клиентом

Все переменные в предлагаемой модели на основе GPSS имеют размерность в условных единицах. Переменная Wait отражает среднее значение временного промежутка для экспоненциального закона, который используется при генерации транзактов в модель. Эти транзакты отождествляются с клиентскими запросами на соединение. Переменные adv1, adv2, adv3, adv4, adv5, adv6 отражают временные интервалы средних значений частей, рассмотренных в предыдущем пункте. Переменная Emk отракжает максимальное количество пользователей для сервера, а переменная Zap – количество пользователей на сервере, которое может занять один клиент (для данного случая каждому клиенту должен ставится в соответствие один пользователь на сервере). ServCl – многоканальное устройство, размер которого должен быть равен переменной Zap. ServCl является имитацией программного сервера, входящего в состав лицензионного и принимающего параллельно запросы на соединение от клиентов. Kont1 – условие, по которому определяется есть ли место на сервере. Kont2 – условие проверки на то, что МКУ пусто.

Блок GENERATE через заданный промежуток времени генерирует транзакт. Следующий за ним блок TEST проверяет МКУ ServCl на исправность (доступность) и наличия в нем свободных каналов, достаточных для удовлетворения запроса. Если булева переменная Kont1 равна 1, транзакт пропускается и занимает МКУ ServCl. Но перед этим вычисляется и

заносится целое число INT(Emk/Zap) в сохраненную ячейку с именем KolPovt, которое определяет сколько транзактов может одновременно находится в МКУ.

После выхода транзакта из блока ENTER начинает работать сегмент учета номеров транзактов, занявших МКУ. Блоком ASSIGN в параметр с именем KolPovt – параметр цикла заносится число, находящееся в сохраняемой ячейке с именем KolPovt. Далее в цикле, тело которого начинается с блока TEST с меткой Met5 и заканчивается блоком LOOP с меткой Met4, находится свободное место в списке для записи номера транзакта. Свободное место определяется блоком TEST как равенство нулю значения какой-либо одной из сохраняемых ячеек по числу транзактов, одновременно находящихся в МКУ. Такая ячейка всегда есть и блоком SAVEVALUE в нее записывается номер занявшего МКУ транзакта. А так как сохраняемых ячеек, значения которых равны нулю, может быть несколько (особенно в начале работы модели), после записи номера транзакта осуществляется выход из цикла.

После этого для транзакта определяется путь, по которому он будет идти дальше. Для этого блоками TRANSFER определяется направление, в котором пойдет транзакт. Всего направлений три: транзакт является клиентским запросом, содержащим не нужные данные (Met14), транзакт является клиентским запросом на задание лицензии (Met15) или транзакт является клиентским запросом на проверку своей лицензии (Met16). В случаях, когда транзакт определяется как запрос клиента, не содержащий не нужных данных, обработка его делится условно на две основные части: работа с клиентом и работа с БД. При работе с БД ее сначала необходимо заблокировать для монопольного доступа. Для этого используются блоки SEIZE и RELEASE для блокирования и разблокирования монопольного доступа соответственно переменной Database, которая ассоциируется с доступом к базе данных.

После обслуживания транзакт освобождает МКУ ServCl, зайдя в блок LEAVE.

Теперь рассмотрим работу модели при возникновении неисправности МКУ. После генерации транзакта блоком GENERATE определяется, может ли он инициировать неисправность при помощи блока TRANSFER. В этом блоке транзакт с вероятностью 0.1% считается способным вызвать неисправность. Транзакт, инициирующий неисправность, входит в блок SUNAVAIL, который переводит МКУ ServCl в недоступное (неисправное) состояние.

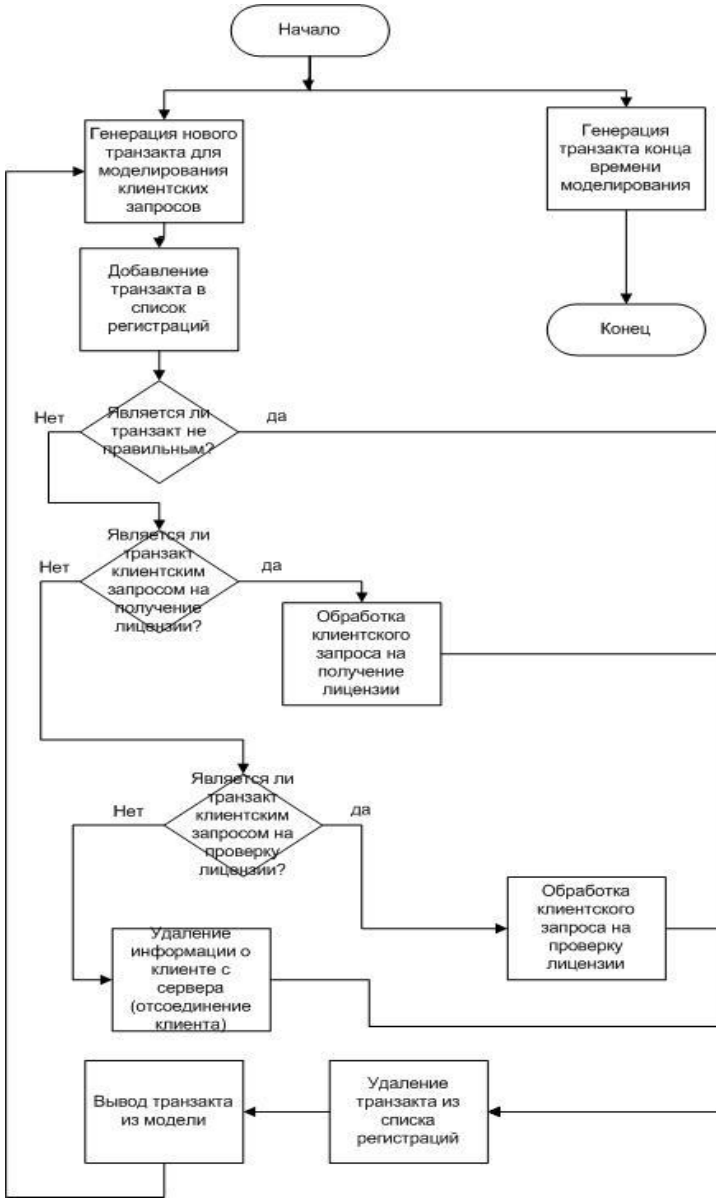

Рисунок 3 – Схема процесса выполнения модели

Далее реализуется удаление транзактов из МКУ. Для этого в цикле, организованном с помощью блока LOOP, просматривается список номеров транзактов, занявших МКУ. Если значение какой-либо сохраняемой ячейки не равно 0, значит, в ней записан номер транзакта, находящегося в данный момент в МКУ. Блоком DISPLACE этот транзакт перемещается к блоку LEAVE с меткой Met3, освобождает МКУ и выводится из модели.

После удаления из МКУ всех транзактов, транзакт-инициатор неисправности входит в следующий за блоком LOOP с меткой Met11 блок TEST. Так как ServCl пусто (булева переменная Kont2 равна нулю), блок TEST пропускает его и начинается имитация восстановления работоспособности МКУ.

Последний блок GENERATE определяет время моделирования.

На рисунке 3 показывается схема выполнения модели.

Результаты моделирования. После выполнения модели выдается стандартный отчет системы GPSS, который включает:

ENTRIES – количество раз, когда устройство было занято или занято с прерыванием начала моделирования или после последнего выполнения оператора RESET или CLEAR (1178913);

UTIL. – коэффициент использования, доля времени моделирования, в течении которого устройство было занято (0.054);

AVE. TIME – среднее время занятия устройства одним транзактом в течении времени моделирования или после выполнения оператора RESET или CLEAR (22875.447);

CAP. – емкость памяти, заданная оператором STORAGE (500);

REM. – число единиц памяти, свободных в конце моделирования (500);

ENTRIES – количество входов в память за период моделирования (999698).

Выводы. На основе системы GPSS World разработана имитационная модель процесса функционирования лицензионного программного обеспечения. Работоспособность модели показывается на тестовом примере. Модель позволяет оценить число пользовательских запросов, которые способен обработать сервер для изначально заданных пропускной способности и затратах временных ресурсов на выполнение элементарных операций по обслуживанию клиентов.

Список литературы: ***1.*** *Томашевский В.Н., Жданова Е.Г.* "Имитационное моделирование средствами GPSS/PC." - К.: ІЗМН, "ВІПОЛ", 1998. - 123 с. ***2.*** *Сытник В.Ф., Орленко Н.С.* "Имитационное моделирование: Учебн. пособие" - К.: КНЕУ, 1998. - 232 с. ***3.*** *Норенков И.П.* Разработка САПР.- М, МГТУ им.Баумана,1994. ***4.*** *Моли Б.* Unix/Linux: теория и практика программирования. Пер. с англ. – М.: КУДИЦ-ОБРАЗ, 2004 – 576 с. ***5.*** www.modbus.org/ ***6.*** *Архангельский А.Я.* Разработка прикладных программ для Windows в C++Builder6.0 – М.: ЗАО "Издательство БИНОМ", 2000. – 1054 с.

Поступила в редколлегию 05.03.01